

## **IN THE CLAIMS:**

1. (Currently amended) A method for editing text used in a user interface of a computer program, comprising the steps of:
  - monitoring execution of the computer program to identify events associated with one or more graphical user interface (GUI) components;
  - generating, during execution of the computer program, contextual information for the identified events, wherein the contextual information provides information defining a context in which text associated with the one or more GUI components is presented;
  - ~~receiving contextual information describing a visual context for the computer program's generation of the text;~~
  - receiving the text ~~from~~ in a first user interface text storage format;
  - combining the contextual information with the text to form a visual representation of a display in the one or more graphical user interface components; and
  - displaying the visual representation; and
  - providing an editor for editing the text in the displayed visual representation.
2. (Currently amended) The method of claim 1, further comprising the step of:
  - receiving editing instructions to edit the text via the editor; and
  - based on the editing instructions, generating new text for use with the one or more graphical user interface components.
3. (Original) The method of claim 2, further comprising the step of:
  - storing the new text in a second user interface text storage format.
4. (Original) The method of claim 1, wherein the first user interface text storage format is a catalog file.
5. (Original) The method of claim 1, wherein the first user interface text storage format is a resource bundle.

6. (Original) The method of claim 1, wherein the contextual information includes graphical user interface (GUI) rendering information.
7. (Original) The method of claim 1, wherein the computer program is written in Java.
8. (Currently amended) A method for generating an editable representation of a user interface in an executing computer program, comprising the steps of:
  - detecting when the executing computer program generates a user interface feature; and
  - recording a description of the user interface feature, during execution of the computer program, wherein the description includes information regarding the rendering of the user interface feature;
  - associating the description with an item of text in a user interface text storage format;
  - replaying generation of the user interface feature based on the description as associated with the item of text; and
  - providing an editor for editing the item of text during replaying of the generation of the user interface.
9. (Canceled)
10. (Canceled)
11. (Original) The method of claim 8, wherein the user interface is a graphical user interface (GUI).
12. (Original) The method of claim 8, wherein the user interface text storage format is a resource bundle.

13. (Original) The method of claim 8, wherein the executing computer program is written in Java.

14. (Currently amended) A method for editing text used in a user interface of a computer program, comprising the steps of:

receiving a sequential record of system events that occurred during an execution session of the computer program;

executing an executable of the computer program, wherein the executable contains an introspective editor; ~~and~~

reproducing the system events from the sequential record to control execution of the executable; and

receiving edit instructions, via the introspective editor, during reproducing of the system events, to thereby edit text used in a user interface presented by execution of the executable of the computer program during reproduction of the system event.

15. (Original) The method of claim 14, further comprising the step of:  
responsive to a user input, suspending execution of the reproducing step.

16. (Original) The method of claim 15, further comprising the step of:  
responsive to a second user input, resuming the execution of the reproducing step.

17. (Original) The method of claim 14, wherein the system events include at least one of a keystroke, a mouse click, a mouse double-click, and a mouse drag.

18. (Currently amended) A computer program product, in a computer-readable medium, for editing text used in a user interface of a computer program, comprising instructions for:

monitoring execution of the computer program to identify events associated with one or more graphical user interface (GUI) components;

generating, during execution of the computer program, contextual information for the identified events, wherein the contextual information provides information defining a context in which text associated with the one or more GUI components is presented;

~~receiving contextual information describing a visual context for the computer program's generation of the text;~~

receiving the text ~~from~~ in a first user interface text storage format;

combining the contextual information with the text to form a visual representation of a ~~display in the~~ one or more graphical user interface components; ~~and~~

displaying the visual representation; and

providing an editor for editing the text in the displayed visual representation.

19. (Currently amended) The computer program product of claim 18, further comprising instructions for:

receiving editing instructions to edit the text via the editor; and

based on the editing instructions, generating new text for use with the one or more graphical user interface components.

20. (Original) The computer program product of claim 19, further comprising instructions for:

storing the new text in a second user interface text storage format.

21. (Original) The computer program product of claim 18, wherein the first user interface text storage format is a catalog file.

22. (Original) The computer program product of claim 18, wherein the first user interface text storage format is a resource bundle.

23. (Original) The computer program product of claim 18, wherein the contextual information includes graphical user interface (GUI) rendering information.

24. (Original) The computer program product of claim 18, wherein the computer program is written in Java.

25. (Currently amended) A computer program product, in a computer-readable medium, for generating an editable representation of a user interface in an executing computer program, further comprising instructions for:

detecting when the executing computer program generates a user interface feature; ~~and~~

recording a description of the user interface feature, during execution of the computer program, wherein the description includes information regarding the rendering of the user interface feature;

associating the description with an item of text in a user interface text storage format;

replaying generation of the user interface feature based on the description as associated with the item of text; and

providing an editor for editing the item of text during replaying of the generation of the user interface.

26. (Canceled)

27. (Canceled)

28. (Original) The computer program product of claim 25, wherein the user interface is a graphical user interface (GUI).

29. (Original) The computer program product of claim 25, wherein the user interface text storage format is a resource bundle.

30. (Original) The computer program product of claim 25, wherein the executing computer program is written in Java.

31. (Currently amended) A computer program product, in a computer-readable medium, for editing text used in a user interface of a computer program, further comprising instructions for:

receiving a sequential record of system events that occurred during an execution session of the computer program;

executing an executable of the computer program, wherein the executable contains an introspective editor; ~~and~~

reproducing the system events from the sequential record to control execution of the executable; and

receiving edit instructions, via the introspective editor, during reproducing of the system events, to thereby edit text used in a user interface presented by execution of the executable of the computer program during reproduction of the system event.

32. (Original) The computer program product of claim 31, further comprising instructions for:

responsive to a user input, suspending execution of the instructions for reproducing.

33. (Original) The computer program product of claim 32, further comprising instructions for:

responsive to a second user input, resuming the execution of the instructions for reproducing.

34. (Original) The computer program product of claim 31, wherein the system events include at least one of a keystroke, a mouse click, a mouse double-click, and a mouse drag.

35. (Currently amended) A data processing system executing a program that displays text, the data processing system comprising:

a bus system;

a memory connected to the bus system, wherein the memory includes a set of instructions; and

a processing unit, wherein the processing unit includes at least one processor, wherein the processing unit executes the set of instructions to perform the acts of:

monitoring execution of the computer program to identify events associated with one or more graphical user interface (GUI) components;

generating, during execution of the computer program, contextual information for the identified events, wherein the contextual information provides information defining a context in which text associated with the one or more GUI components is presented;

~~receiving contextual information describing a visual context for the computer program's generation of the text;~~

receiving the text ~~from~~ in a first user interface text storage format;

combining the contextual information with the text to form a visual representation of ~~a display in the~~ one or more graphical user interface components; ~~and~~

displaying the visual representation; and

providing an editor for editing the text in the displayed visual representation.

36. (Currently amended) The data processing system of claim 35, wherein the processing unit executes the set of instructions to perform the additional acts of:

receiving editing instructions to edit the text via the editor; and

based on the editing instructions, generating new text for use with the one or more graphical user interface components.

37. (Original) The data processing system of claim 36, wherein the data processing system executes the set of instructions to perform the additional act of:

storing the new text in a second user interface text storage format.

38. (Original) The data processing system of claim 35, wherein the first user interface text storage format is a catalog file.

39. (Original) The data processing system of claim 35, wherein the first user interface text storage format is a resource bundle.
40. (Original) The data processing system of claim 35, wherein the contextual information includes graphical user interface (GUI) rendering information.
41. (Original) The data processing system of claim 35, wherein the program is written in Java.
42. (Currently amended) A data processing system to generate an editable representation of a user interface in an executing program, the data processing system comprising:
- a bus system;
  - a memory connected to the bus system, wherein the memory includes a set of instructions; and
  - a processing unit, wherein the processing unit includes at least one processor, wherein the processing unit executes the set of instructions to perform the acts of:
    - detecting when the executing computer program generates a user interface feature; ~~and~~
    - recording a description of the user interface feature, during execution of the computer program, wherein the description includes information regarding the rendering of the user interface feature;
    - associating the description with an item of text in a user interface text storage format;
    - replaying generation of the user interface feature based on the description as associated with the item of text; and
    - providing an editor for editing the item of text during replaying of the generation of the user interface.
43. (Canceled)



44. (Canceled)

45. (Original) The data processing system of claim 42, wherein the user interface is a graphical user interface (GUI).

46. (Original) The data processing system of claim 42, wherein the user interface text storage format is a resource bundle.

47. (Original) The data processing system of claim 42, wherein the executing program is written in Java.

48. (Currently amended) A data processing system to edit text used in a user interface of a program, the data processing system comprising:

- a bus system;

- a memory connected to the bus system, wherein the memory includes a set of instructions; and

- a processing unit, wherein the processing unit includes at least one processor, wherein the processing unit executes the set of instructions to perform the acts of:

- receiving a sequential record of system events that occurred during an execution session of the computer program;

- executing an executable of the computer program, wherein the executable contains an introspective editor; ~~and~~

- reproducing the system events from the sequential record to control execution of the executable; and

- receiving edit instructions, via the introspective editor, during reproducing of the system events, to thereby edit text used in a user interface presented by execution of the executable of the computer program during reproduction of the system event.

49. (Original) The data processing system of claim 48, wherein the data processing system executes the set of instructions to perform the additional act of:

responsive to a user input, suspending execution of the instructions for reproducing.

50. (Original) The data processing system of claim 49, wherein the data processing system executes the set of instructions to perform the additional act of:

responsive to a second user input, resuming the execution of the instructions for reproducing.

51. (Original) The data processing system of claim 48, wherein the system events include at least one of a keystroke, a mouse click, a mouse double-click, and a mouse drag.

52. (New) The method of claim 1, wherein displaying the visual representation includes providing the contextual information and the text to a context interpreter which generates the visual representation as a recreation of the text, in the context described by the contextual information, without using the computer program to generate the visual representation.

53. (New) The method of claim 1, wherein displaying the visual representation includes providing the contextual information and text to a scripting shell, wherein the scripting shell replays an execution session of the computer program, and wherein providing an editor for editing the text in the displayed visual representation includes providing an introspective editor during replaying of the execution session of the computer program.

54. (New) The method of claim 53, wherein the introspective editor is provided during replaying of the execution session of the computer program in response to user input halting replaying of the execution session.

55. (New) The method of claim 8, wherein replaying generation of the user interface feature includes providing the description and item of text to a scripting shell, wherein

the scripting shell replays an execution session of the computer program, and wherein providing an editor for editing the item of text includes providing an introspective editor during replaying of the execution session of the computer program.

56. (New) The method of claim 55, wherein the introspective editor is provided during replaying of the execution session in response to user input halting replaying of the execution session.